

Spécifications du Mini-Projet : Gestionnaire de Rendez-vous Hospitaliers (GRH)

Technologies : Java + MongoDB (NoSQL)

1. Objectif du Projet

Développer une application intitulée **Gestionnaire de Rendez-vous Hospitaliers (GRH)** permettant aux patients de **réserver, modifier et annuler des rendez-vous** avec des médecins.

Le système doit également permettre aux **administrateurs** de gérer les **médecins, les patients et les plannings de rendez-vous** à l'aide d'une base de données **MongoDB** et d'un **backend développé en Java** (console, interface graphique ou API REST).

2. Exigences du Système

Exigences Fonctionnelles

A. Gestion des Patients

- Enregistrer un nouveau patient avec les informations suivantes : nom, date de naissance, sexe, téléphone, email, adresse et identifiant.
- Consulter, modifier ou supprimer les informations d'un patient.
- Rechercher un patient par nom ou identifiant.

B. Gestion des Médecins

- Ajouter, modifier ou supprimer un médecin.
- Enregistrer les détails du médecin : nom, spécialité, coordonnées et jours/heures de travail.
- Rechercher un médecin par spécialité ou par nom.

C. Gestion des Rendez-vous

- Réserver un nouveau rendez-vous (patient + médecin + date + heure).
- Vérifier les créneaux disponibles avant la réservation.
- Mettre à jour ou annuler un rendez-vous.
- Afficher tous les rendez-vous à venir par date ou par médecin.
- Marquer automatiquement les rendez-vous passés comme « Terminés ».

D. Rapports et Requêtes

- Lister tous les rendez-vous d'un jour donné.
- Compter le nombre de rendez-vous par médecin ou par spécialité.
- Identifier les patients ayant plusieurs visites récentes.
- Afficher les créneaux horaires disponibles pour un médecin à une date donnée.

Exigences Non Fonctionnelles

- **Langage de programmation :** Java
- **Base de données :** MongoDB
- **Bibliothèques / Frameworks :**

- Pilote Java MongoDB
- *Optionnel* : Spring Boot pour la création d'API REST
- **Interface :**
 - Interface graphique
 - *Optionnel* : API REST (Spring Boot)
- **Performance** : Les opérations CRUD.
- **Sécurité** : Validation des entrées ; éviter les doubles réservations pour un même médecin et créneau.

4. Modélisation de la Base de Données (Schéma MongoDB)

Collection : patients

```
{
  "_id": "ObjectId",
  "patient_id": "P1001",
  "name": "Ali Ben Omar",
  "dob": "1990-08-21",
  "gender": "Homme",
  "phone": "+212600123456",
  "email": "ali.benomar@example.com",
  "address": "Casablanca, Maroc"
}
```

Collection : doctors

```
{
  "_id": "ObjectId",
  "doctor_id": "D2001",
  "name": "Dr. Laila Kharraz",
  "specialization": "Dermatologie",
  "email": "lkharraz@clinic.ma",
  "phone": "+212612345678",
  "working_days": ["Lundi", "Mardi", "Jeudi"]
}
```

Collection : appointments

```
{
  "_id": "ObjectId",
  "appointment_id": "A3001",
  "patient_id": "P1001",
  "doctor_id": "D2001",
  "date": "2025-11-10",
  "time": "10:30",
  "status": "Planifié",
  "remarks": "Contrôle dermatologique"
}
```

5. Fonctions Principales en Java

Gestion des Patients

```
ajouterPatient();  
modifierPatient();  
supprimerPatient();  
rechercherPatientParId();  
listerTousLesPatients();
```

Gestion des Médecins

```
ajouterMedecin();  
modifierMedecin();  
supprimerMedecin();  
rechercherMedecinParSpecialite();  
listerTousLesMedecins();
```

Gestion des Rendez-vous

```
reserverRendezVous();  
modifierRendezVous();  
annulerRendezVous();  
afficherRendezVousParMedecin();  
afficherRendezVousParDate();  
trouverCreneauxDisponibles();
```

6. Fonctionnalités Avancées

- API REST (/patients, /doctors, /appointments).
- Notifications par email/SMS (mock).
- Rapports automatiques (PDF, Excel).
- Tableau de bord médecin : rendez-vous du jour et statistiques.
- Portail patient : historique des rendez-vous.

7. Livrables

1. Code source Java complet.
2. Export MongoDB (JSON) avec données d'exemple.
3. Rapport de projet (3 à 4 pages) comprenant :
 - Objectifs et problématique
 - Architecture et conception
 - Schéma de base de données
 - Captures d'écran / exemples d'exécution
 - Conclusion et perspectives.